Modeling Character in Tweets

April 2017

John Martin
College of Computer and Information Science
Northeastern University
Boston, MA 02115
martin.john@husky.neu.edu

Abstract

This project considers the task of modeling the individual character of users on Twitter. Sentiment analysis with Naïve Bayes and user cross entropy is calculated in various configurations to model user character and determine if particular tweets are in or out of character for a given user. For various reasons, however, the performance of the models met downfalls. The paper concludes with a discussion of improvements for future work.

1 Introduction

People receive information more rapidly, and in greater volume, than ever before. Social media has allowed for individuals to consume this information, but also to respond to it. These responses, or reactions, might come without much forethought. Indeed, politicians and corporate leaders walk back their initial statements made on social media all the time. But are these responses truly out of character? Do people behave in a consistent manner on the internet? This research aims to use methods of natural language processing to model online reactions and interactions, exposing when—or if—individuals have expressed something out of the ordinary. In essence, the goal is to show how like or unlike *themselves* individuals have acted.

2 Previous Work

Much related research has been conducted within sociology and psychology to uncover the human nature of this topic. People are generally said to act unlike themselves in elevated situations, such as emergencies or situations involving significantly increased emotion, be it anger, depression or euphoria. Many psychologists believe this reactionary behavior is a truer, albeit unrefined, representation of the sincere feelings or beliefs of the individual. It is this deviation, expressed in social media, that is under investigation in this research.

Little to no technical work has been done specifically on this task, although Twitter is commonly used for natural language processing research and sentiment analysis.

Research that does exist on the topic focuses on broader reactionary influences in social media. For instance, Rebekah Giordano published a case study[1] of how Twitter users behaved after a major event—the Boston Marathon bombing. The research focuses on questions of which information sources were most engaged and the frequency of tweets related to the event. Other than training a Naïve Bayes model on tweet sentiment, the research chronicled in this paper focuses entirely on individual users and their individual behavior.

3 Methodology

For much of this work, "controversial" tweets were chosen manually, while software analyzed replies and the repliers' profiles, or "timelines." The general procedure for this research was (1) select a controversial subject and find a popular tweet for it, (2) consume the replies to that tweet, (3) consume the timelines for each user who responded and build models based on those profiles, (4) compare the original set of reply tweets to their authors' timeline models. Unfortunately, Twitter has a rather limited API, and some creative workarounds were needed to get data. It should be mentioned that certain tools (twarc[2], tweepy[3]) were used to interacting with Twitter data. All tweets collected were authored by public Twitter accounts, and the names and usernames of users other than politicians have been redacted from this report.

The two major machine learning methods used to analyze tweet reactions were (1) Naïve Bayes and (2) cross entropy.

3.1 Naïve Bayes sentiment

At first, Naïve Bayes models were generated as described above, using only users' timelines as the corpus and deploying a leave-one-out strategy for classifying the reactionary tweet in question. However, because there was only one class in the Naïve Bayes model (that of the user in question), there was only one log likelihood to consider, and no pigeonholing could be performed. This was a foolish oversight. The second problem encountered with this strategy was insufficient timeline information from users. The Twitter API only allows for the collection of the most recent 3.2K tweets from a user, which is too small a corpus to reliably train a Bayesian classifier on. The solution to this was to find online corpora of Twitter timelines of greater size[4]. One in particular that was used is that of Donald Trump (`@realDonaldTrump`). This increased the corpus size by roughly one order of magnitude, which is significant. The tweet corpus of Donald Trump and other political figures was used for the rest of the experiments. This was not done because the experiments are meant to quantify about politicians specifically, but because politicians happen to have their Twitter data being archived for use.

The second, more fruitful Naïve Bayes experiment involved using sentiment analysis to classify tweets. This experiment is based on the assumption that certain hashtags used in tweets will generally correlate to

[1] http://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1001&context=journalismprojects

[2] https://github.com/docnow/twarc

[3] https://github.com/tweepy/tweepy

[4] https://github.com/bpb27/trump_tweet_data_archive

certain sentiment. Large corpora of positive and negative tweets were collected from Twitter with select hashtags (Figure 1), and Naïve Bayes models were built from those collections. Then, similar to before, the timelines of users were collected and an average sentiment ratio was generated for the user by classifying each tweet into either the positive or negative sentiment class. Finally, the reactionary tweet in question was classified, demonstrating if the sentiment of the reaction was out of the ordinary for the given user.

Public hashtags collected for the positive and negative sentiment classes

Positive	Negative
#blessed	#angry
#ecstatic	#crushed
#excited	#depressed
#happy	#frustrated
#overjoyed	#furious
	#lonely
	#sad

Figure 1

All tweet text was converted to lowercase, stripped of punctuation, handles (names of users; i.e. @POTUS) were replaced with a special NAME token, and hashtags (i.e. #throwbackthursday) were left with their pound character # to preserve their differentiation from other tokens. Finally, all words appearing with a frequency less than five were removed from the corpus.

3.2 Cross entropy

Perhaps a more appropriate way of analyzing how unpredictable a reactionary tweet is, is by modeling that tweet's cross entropy with that of the corpus of the user's timeline. For this experiment, both trigram and bigram models were built to estimate the next word (note: not character) of tweet text given the preceding word or two. All tweet text was manipulated similarly as before, converting to lowercase, removing punctuation, replacing handle tokens, and preserving hashtags. Each tweet was considered individually and the initial and final bigrams and trigrams—those containing theoretical start and end tokens—for any given tweet were ignored for simplicity.

After analyzing manually-chosen reactionary tweets, a second experiment was conducted on users to find the most usual and unusual tweets. That is, the cross entropy of all tweets of a user was calculated and ordered from highest to lowest entropy. This allows the model to express what it believes to be the tweets which are most unlike their author. This can be used to retroactively discover which subjects elicit atypical reactions from a particular user.

Cross entropy was calculated as the number of bits required to encode each word using the following formula where *n* is the number of words in the tweet and *p(w)* is the probability of the bigram or trigram in the corpus with lambda smoothing of 0.1:[5]

$$-\frac{1}{n}\sum_{w_1 \dots w_n} p(w_1 \dots w_n)\, log_2\, p(w_1 \dots w_n)$$

4 Results

4.1 Naïve Bayes sentiment

[5] https://courses.engr.illinois.edu/cs498jh/Slides/Lecture04.pdf

As can be seen in the table below, of 30,869 tweets by Trump categorized, 20,164 were negative. This means Trump is rather exceptionally negative on Twitter, having a positive tweet frequency of roughly 0.35. This means one would generally expect Trump's tweets to be negative, and positive tweets would be unusual for his character. One would expect the opposite for his daughter, `ivankatrump`. Unfortunately, many users are not skewed very heavily one way or the other in terms of sentiment, so it is less useful to make a statement about some particular reactionary tweet for those users. This is a shortcoming of this approach.

Counts of positively and negatively classified tweets per user

user	num pos	num neg	pos freq
realDonaldTrump	10705	20164	0.35
marcorubio	27324	24768	0.52
tedcruz	61524	64629	0.49
senjohnmccain	47583	55332	0.46
sarahpalinusa	20637	19611	0.51
citizens_united	27720	54144	0.34
ivankatrump	8763	2500	0.78

Figure 2

Because of this shortcoming, sentiment categorization of reactionary tweets is only as interesting as the relative sentiment behavior of the user. Even in cases such as Donald and Ivanka Trump where one can expect a certain sentiment more often, many tweets have mostly unsurprising categorization. For example, both Donald and Ivanka have tweets that read "Happy Easter", which is categorized by the sentiment model as positive. For Donald, that tweet is unusual, and for Ivanka it is considered usual. In cases like these, the experiment has mostly said something interesting about the character of the particular user, and not much interesting about any particular tweet by a user. Specifically, it successfully describes the average attitude of a user on Twitter.

Though the situations may be limited, this model is capable of producing meaningful results. Ivanka, who rarely makes political statements on Twitter, tweeted a statement supporting her father's choice to strike Syria in April 2017, a tweet that was a direct response to a tweet from Donald. The model categorized the tweet as having negative sentiment, which is out of character for Ivanka's timeline corpus.

@ivankatrump tweet classified as out of sentiment character

```
"the times we are living in call
for difficult decisions proud of
my father for refusing to accept
these horrendous crimes against
humanity"
```

Figure 3

4.2 Cross entropy

When considering the cross entropy of any individual tweet, it is helpful to understand what entropies rating the model is generally assigning to most tweets. The following graphs—**Error! Reference source not found.** and Figure 5—show the counts of entropy ratings for a large corpus—30K tweets in Figure 4—and a small corpus—3.2K tweets in Figure 5. As can be seen, both the trigram and bigram models for both corpus sizes gave nearly all tweets a rating between 0.0 and 0.4, or low predictability. This suggests that the corpora may not have been large enough, which is discussed more thoroughly below in Section 5.1.

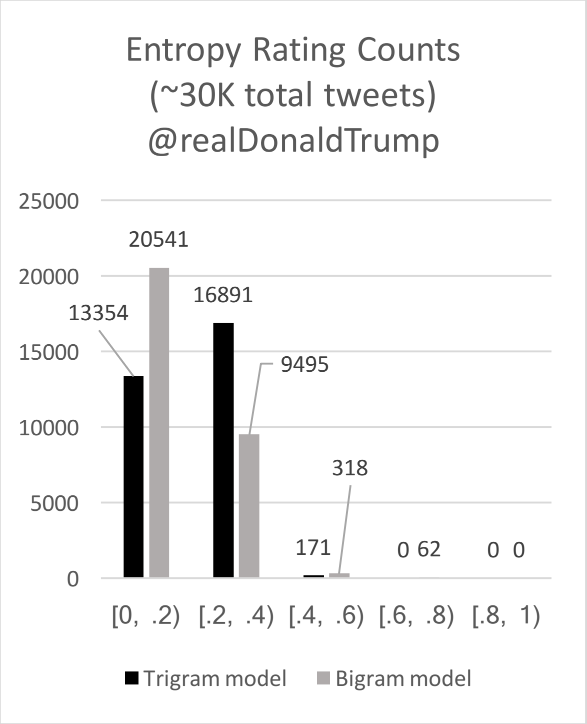

Figure 4

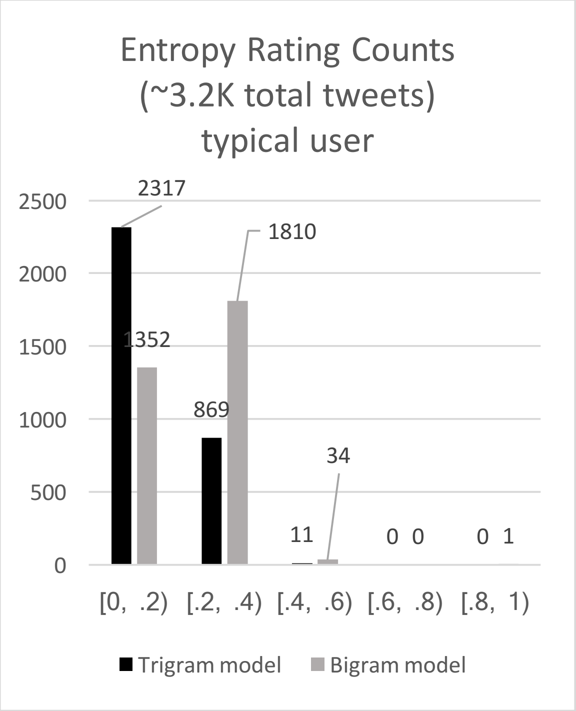

Figure 5

To better understand why entropy distributions were this way, it is helpful to look at specific tweets that were categorized:

Examples of the most predictable, or least entropic, tweets

`@realDonaldTrump`, *trigram modeling, predictable:*

- “repeal and replace obamacare”
- “china owes us money LINK #trumpvolg”
- “reporter should resign LINK”
- “happy easter to everyone”
- “NAME good luck”
- “make america great again”

Figure 6

`@realDonaldTrump`, *bigram modeling, predictable:*

- “NAME wow very nice”
- “make america great again”
- “NAME thanks have fun”
- “NAME NAME yes”
- “NAME NAME great”
- “NAME thanks my honor”

Figure 7

Examples of the least predictable, or most entropic, tweets

`@realDonaldTrump`, *trigram modeling, unpredictable:*

- “NAME NAME agree 100 percent”
- “on ruining scotland beauty with ugly costly wind turbines”
- “NAME apology accepted”
- “failure defeats losers failure inspires winners robert t kiyosaki NAME”
- “chance favors the prepared mind louis pasteur"

Figure 5

`@realDonaldTrump`, *bigram modeling, unpredictable:*

- “NAME thsnk you”
- “NAME be concise”
- “NAME sikorsky 76”
- “NAME night loser”
- “NAME no #darell does”

Figure 6

The results of calculating the cross entropy of all user tweets against the user’s corpus shows that the model seems to do well when dealing with relatively low entropy, or predictable tweets. However, it suffers when dealing with tweets of high entropy. For example, the model found the tweet with a typo, “thsnk you”, to have high entropy, but of course the real meaning of the tweet, “thank you”, would have low entropy. This stems from the informal nature of Twitter. Misspellings are often not corrected by authors. Additionally, the model cannot account for identical meanings in purposeful misspelling that are common on Twitter (i.e. “you” and “u”).

Additionally, names like “Louis Pasteur” or “Robert T. Kiyosaki” inflate entropy in the model because they are rare by nature of being proper nouns. As can be seen in the good low entropy examples, replacing handles with a special “NAME” token kept predictability high for these cases in a helpful way. Possible modifications to this are discussed below in Second 5.4.

5 Discussion

5.1 Size of data set

Corpus size of Twitter user timelines was one of the bigger issues for this project. The 3,200 tweet limit on any individual user was obviously too shallow to train any meaningful models. Even archived timelines of politicians that that had as many as 35,000 tweets did not amount to impressively expansive corpora, as could be seen in the entropy classes in Figure 4 and Figure 5. It is reasonable to question the validity of these results for the small datasets available from Twitter users. There is not an obvious solution to this without adjusting the experiment, because users will only tweet so much.

It should be noted also that much of a user’s tweets are actually “retweets”, which means the text of the tweet was not drafted by that user. In these experiments, all tweets, including retweets, were used for a user in order to increase corpora sizes. It can be argued that any retweets also represent the thoughts or behavior of a user, even if it was not drafted by that user. However, it may also be the case that retweets diminished the authentic character that could be produced by training models on text drafted exclusively by the user in question.

Data set size should not have been a problem for the Bayesian sentiment models, though. There are few limits on the Twitter search API, which is how the data was obtained for the sentiment analysis, so the set was sufficiently large. To be explicit, the positive and negative sentiment corpora contained nearly one million tweets. It is reasonable to believe the model was fairly well trained at this size.

5.2 Average users in sentiment categorization

Though data set size was likely not an issue for the sentiment models, there were other issues. Many users on Twitter were categorized to be quite average by the model, meaning roughly half of their tweets were "positive" and half were "negative". This was problematic for the hypothesis that something out of the ordinary might be said about some particular tweet, for if nothing unusual can be said about the normal sentiment of the user, little can be said about any particular tweet by that user.

One proposed solution to this is combining all the text of all tweets of a user when determining their sentiment. The model used considered each tweet independently and took an average of the binary results—each tweet was counted as being wholly positive or wholly negative. If instead user sentiment is determined by considered all tweets concatenated, there may be a more precise rating generated.

A second possible solution to this problem is adding more dimensions of sentiment. Other Twitter sentiment research has found success in more dimensions than one.[6] If more minutia of emotion could be accurately modeled, one would likely be able to derive a more complete sentiment profile for a user, and thus have more dimensions to judge a particular tweet on.

5.3 Selection of sentiment data

As expressed above in Section 3.1, sentiment data was selected by consuming the text of tweets containing certain hashtags (Figure 1). The possibly bold assumption being made here is that the overall sentiment of all, or at least most, such tweets was in fact positive or negative. While the results of the sentiment classification suggest this is largely true, it is not a particularly robust method for defining sentiment. Much work has been done in Twitter sentiment and sentiment analysis in general. Other implementations could lead to better results than exist here, but Naïve Bayes classifiers have proven to work well so this likely was a good model, if not necessarily a true model.[7]

For instance, the example tweet mentioned above in Figure 3 accurately classifies Ivanka's tweet as having out-of-character sentiment. However, the sentiment classification of that tweet, negative, might not actually be a true colloquial categorization that a human or more sophisticated model would assign. Indeed, it seems rather positive. But since the model is consistently judging all tweets, the truth of the sentiment does not matter for determining what is out of character. One can simply think of the classes as arbitrary binaries, and the validity of tweet classification on those classes holds.

5.4 Issues of cross entropy

The results of cross entropy of are more satisfactory as a method for classifying the character of a user. Indeed, it intuitively seems truer to the question of how predictable a particular tweet is to a class tweets. Unfortunately, the corpus size and informal nature of text on Twitter led to issues in classifying tweets. As discussed above, typos and proper nouns disrupted the model significantly.

[6] https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/

[7] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan

In a future experiment, it may be helpful to first perform some type of manual or automatic clustering on words before training the cross entropy model. This could feasibly collapse words like “you” and “u” into one meaning, and collapse typos like “thsnk” into a more informative meaning.

Additionally, proper nouns that are mentioned once or never by the user should ideally not be considered when calculating entropy. Also discussed above, user handles were replaced by a special token because the specific individual is not information that is desired in the model, but just the fact that *some* person is mentioned. The handle syntax (`@`) on Twitter allowed for an obvious method of doing this replacement, but a much more sophisticated method would need to be used to catch other proper nouns and names.

Similarly, for simplicity in this experiment, links were left unmodified in the model, and not replaced by any special token. Given that links are almost always unique, this certainly generated noise in the model that ideally would not be included. Developing a method of redacting links could be simpler than that of a proper noun identifier, and thus could be a fruitful improvement to this approach.

6 Conclusion

Overall, this project is a fair start to exploring the behavior, or character, of individuals online. Neither of the two machine learning methods used for classifying tweet text, Naïve Bayes and cross entropy, were fruitless in saying something about the character of Twitter users. As discussed, this project met several issues, including corpus size. For some users—those who simply do not use social media often—these models are ineffective because they require larger training corpora than might be available.

However, this project did show that something can be derived about the character of users on social media. Specifically, their average and deviating sentiments, and a predictability rating for individual expressions. Future work can hopefully improve on these methods, and potentially form the foundation for systems that can detect when particular users—perhaps powerful politicians or corporate leaders—engage in social media in uncharacteristic ways.

References

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Association for Computational Linguistics*, pages 79-86.